Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for providing a response to a request for information from a client computing system to a server computing system having an output cache for storing portions of web pages corresponding to at least partial responses to previous requests for information, the method comprising:

receiving, at the server computing system, a request for information from the client computing system;

creating a page object having references to objects on the server computing system in response to the received request for information, wherein creating the page object includes, when the output cache contains ~~a pre-rendered~~ output data of an object referenced by the page object, retrieving the ~~pre-rendered~~ output data of the object referenced by the page object ~~contained in the output cache is retrieved~~ from the output cache and, when the output cache does not contain ~~a pre-rendered~~ output data of an object referenced by the page object, retrieving executable code for the object ~~referenced by the page object~~ not contained in the output cache ~~is retrieved~~ from another source and ~~instantiated~~ instantiating the executable code to create the object referenced by the page object;

inserting ~~the~~ any retrieved ~~pre-rendered~~ output data and ~~the~~ any created objects ~~referenced by the page object~~ as components into a hierarchical tree data model at the server computing system ~~such that each component is linked to at least a prior component~~;

~~rendering~~ processing the components of the hierarchical tree data model at the server computing system to create a ~~rendered~~ renderable page object; and

sending ~~contents of~~ the ~~created rendered~~ renderable page object from the server computing system to the client computing system.

2. (Currently Amended) The method of claim 1, wherein:

the ~~created~~ renderable page object includes a reference to a user control object, the user control object including instructions for obtaining data and an output caching directive for caching output data generated by ~~rendering~~ processing the user control object for the ~~created~~ renderable page,

the step of ~~rendering~~ processing further comprises:

executing instructions of the user control object to obtain the data and the output data; and

storing the output data in the output cache.

3. (Currently Amended) The method of claim 1, wherein the contents of the ~~created rendered~~ renderable page comprises an HTML specification for a web page.

4. (Currently Amended) The method of claim 2, wherein:

the ~~created rendered~~ renderable page includes at least one control; and

the step of inserting a component includes inserting a component corresponding to each respective one of the at least one control~~; and~~

~~the step of rendering the page comprises rendering each one of the components individually~~.

5. (Currently Amended) The method of claim 4, further comprising:

creating the hierarchical tree data model including each of the components and a hierarchical relationship among the components, the data model being used during the step of ~~the rendering~~ processing the page to ~~render~~ facilitate processing each of the components.

6. (Original) The method of claim 2, wherein the output caching directive includes a time duration during which the output data is permitted to reside in the output cache.

7. (Original) The method of claim 6, wherein the output caching directive includes an attribute indicating a condition for varying the output data to be stored in the output cache.

8. (Original) The method of claim 7, wherein the attribute indicates that the output data is to be stored in the output cache according to a type of browser used by the client computing system.

9. (Original) The method of claim 7, wherein the attribute indicates that the output data is to be stored in the output cache according to values of at least one parameter.

10. (Original) The method of claim 1, further comprising providing, on the server computing system, performance counters to monitor output caching performance.

11. (Original) The method of claim 10, wherein the performance counters include:

an output cache hit counter to count a number of requests serviced from the output cache; and

an output cache miss counter to count a number of failed output cache requests.

12. (Original) The method of claim 10, wherein the performance counters include an output cache turnover rate to count a number of additions and removals to the output cache per second.

13. (Original) The method of claim 10, wherein the performance counters include an output cache hit ratio to keep track of a percentage of total requests serviced from the output cache.

14. (Currently Amended) A machine-readable medium having instructions recorded thereon, such that when the instructions are read and executed by a processor in a first computing system connected to a network~~, the computer system functions as a server computer system and the server computer~~ the first computing system performs a method comprising:

receiving, at the ~~server~~ first computing system, a request for information from ~~the client~~ a second computing system;

creating a page object having references to objects on the ~~server~~ first computing system in response to the received request for information, including:

~~when the output cache contains a pre-rendered output data of an object referenced by the page object, the~~ retrieving from an output cache ~~pre-rendered~~ output data of ~~the~~ any object that is referenced by the page object and contained in the output cache ~~is retrieved from the output cache~~ , and

~~when the output cache does not contain a pre-rendered output data of an object referenced by the page object,~~ retrieving from another source executable code for ~~the~~ any object that is referenced by the page object and not contained in the output cache ~~is retrieved~~ ~~from another source~~ and ~~instantiated~~ instantiating the executable code to create the objects ~~object~~ ~~referenced by the page object~~;

inserting the retrieved ~~pre-rendered~~ output data and the created objects ~~referenced by the page object~~ as components into a hierarchical tree data model at the first computing system ~~such that each component is linked to at least a prior component~~;

~~rendering~~ processing the components of the hierarchical tree data model to create ~~a rendered~~ a renderable page at the first computing system; and

sending ~~contents of~~ the created ~~rendered~~ renderable page from the first computing system to the ~~client~~ second computing system.

15. (Currently Amended) The medium of claim 14 wherein:

the created renderable page object includes a reference to a user control object, the user control object including instructions for obtaining data and an output caching directive for caching output data generated by ~~rendering~~ processing the user control object for the created renderable page object,

the step of ~~rendering~~ processing further comprises:

executing instructions of the user control object to obtain the data and the output data; and

storing the output data in the output cache.

16. (Currently Amended) The medium of claim 14, wherein the contents of the created ~~rendered~~ renderable page comprises an HTML specification for a web page.

17. (Currently Amended) The medium of claim 15, wherein;

the created ~~rendered~~ renderable page includes at least one control;

the step of inserting a component includes inserting a component corresponding to each respective one of the at least one control; and

the step of ~~rendering~~ processing the ~~page~~ created objects comprises ~~rendering~~ processing each one of the components individually.

18. (Currently Amended) The medium of claim 17, further comprising:

creating the hierarchical tree data model including each of the components and a hierarchical relationship among the components, the data model being used during the step of ~~rendering~~ processing the page to ~~render~~ process each of the components.

19. (Original) The medium of claim 15, wherein the output caching directive includes a time duration during which the output data is permitted to reside in the output cache.

20. (Original) The medium of claim 19, wherein the output caching directive includes an attribute indicating a condition for varying the output data to be stored in the output cache.

21. (Currently Amended) The medium of claim 20, wherein the attribute indicates that the output data is to be stored in the output cache according to a type of browser used by the ~~client~~ second computing system.

22. (Original) The medium of claim 20, wherein the attribute indicates that the output data is to be stored in the output cache according to values of at least one parameter.

23. (Currently Amended) The medium of claim 14, further comprising providing, on the ~~server~~ first computing system, performance counters to monitor output caching performance.

24. (Original) The medium of claim 23, wherein the performance counters include:

an output cache hit counter to count a number of requests serviced from the output cache; and

an output cache miss counter to count a number of failed output cache requests.

25. (Original) The medium of claim 23, wherein the performance counters include an output cache turnover rate to count a number of additions and removals to the output cache per second.

26. (Original) The medium of claim 23, wherein the performance counters include an output cache hit ratio to keep track of a percentage of total requests serviced from the output cache.

27. (New) A method for providing a response to a request for information from a client computing system to a server computing system having an output cache for storing static portions of web pages, the method comprising:

receiving a request from the client computing system for a web page having a plurality of components, each of the components being either a static component or a dynamic component;

generating the requested web page including:

determining whether an output cache on the server computing system contains any static components of the web page;

retrieving each of the static components contained in the output cache;

creating at the server computing system each of the static components not contained in the output cache by retrieving executable code for each respective component from another source and instantiating the retrieved executable code;

creating at the server computing system each of the dynamic components by retrieving executable code for each respective component from another source and instantiating the retrieved executable code; and assembling the static components and the dynamic components into a hierarchal data model;

generating contents for the web page by processing each of the static components and each of the dynamic components of the hierarchal data model; and

sending the generated contents to the client computing system.